White Paper Report

Report ID: 100846

Application Number: HD5108710

Project Director: Christopher Raphael (craphael@indiana.edu)

Institution: Indiana University, Bloomington

Reporting Period: 9/1/2010-8/31/2011

Report Due: 11/30/2011

Date Submitted: 11/30/2011

Optical Music Recognition on the IMSLP

Prof. Christopher Raphael
Indiana University, Bloomington

November 30, 2011

1 Overview and Goals

The goal of our project is to create an optical music recognition (OMR) system that will transform the images of the International Score Library Project (IMSLP) into *symbolic music representations* — encodings that express music at the note level in a manner analogous to character-encoded text. Symbolic music representations enable music to be automatically searched, transformed, analyzed, and classified, however, very little music is currently available in symbolic form, thus preventing music from fully joining the information revolution of the current century. OMR is the obvious key to developing symbolic music databases since music scores (unlike audio) directly express the core information we seek (discrete pitch and musical time, parts, etc.). Furthermore, with the advent of the IMSLP, our goal is especially timely. The IMSLP is a rapidly growing website, currently containing about 140,000 classical, public domain music scores, mostly in pdf format. Many musicians make daily connect with the library, while its use has become progressively more mainstream — it is quite possible that the IMSLP will be the world's "main" library for public domain music in the future. The IMSLP's large score collection *implores* our music informatics community to harvest this potential gold mine of symbolic data. However, the IMSLP provides an equally important, though less tangible resource. The library is supported by a seemingly altruistic community of musicians whose common goal is the widespread sharing of music. Since the construction of symbolic music libraries will never be completely "solved" through OMR, we hope to leverage the enthusiasm and generous spirit of this community. Specifically, we hope IMSLP community will adopt the open-source software we will create, providing the human-guidance necessary to create high-quality symbolic music scores.

Various OMR efforts have been scattered throughout the last half century, including a great many (mostly short-lived) academic efforts and about a dozen commercial systems. In spite of the many efforts involved, the state of the art is not nearly good enough to address the challenge posed by the IMSLP. While current systems often perform reasonably well in the "laboratory," the realities of "in vivo" experiments (image degradation, imperfect scanning, the variety of music fonts, the inevitable overlap and occlusion of symbols, the long-tailed distribution of possible musical symbols and conventions, etc.) frequently confound these systems. Our approach will significantly extend the state of the art through a combination of three ingredients: better recognition science, sustained effort, and reframing the basic problem, as follows.

Better Recognition Science From a scientific vantage point, existing OMR approaches do not make the obvious connections to the last few decades of research in optical character recognition (OCR) and speech recognition. Both of these parallel fields rely on computational paradigms ideally suited to the *one-dimensional nature* of OCR data (lines of text) and speech (sequence data). While these ideas cannot be adapted wholesale to OMR, they carry a highly relevant core of modeling techniques and associated algorithms. Our initial efforts in OMR have been highly successful in developing core methodology that leverages this existing scientific knowledge. The PI's experience as a former OCR researcher facilitates this process. In addition we have introduced new ideas to deal with the many challenging aspects of OMR not addressed in parallel recognition areas, such as its fundamentally two-dimensional nature.

Sustained Effort An OMR system that tackles the grand challenge presented by the IMSLP will, of course, not be built overnight. In fact, most past OMR approaches have met with limited success partly due to the limited scope of their efforts. In many cases, this is justified by a legitimate focus on recognition science rather than practical results. In contrast, our aim is to *build a working system* that will be adopted in practice by the IMSLP community to accomplish our shared goals. Our project has been pursued with this long-term vision in mind, focusing initial efforts on the foundation that will support much continued work.

Involving Human Input The "good news" for OMR is that music notation begins with a rather simple core of notational conventions for expressing pitch and rhythm that account for the majority of ink on the page. These conventions are, for the most part, consistently followed throughout "common music notation." However, heaped on top of these conventions are several centuries worth of modifications, some common (articulations, bowing, fingering, trills, dynamics, pedaling, arpeggiations, repeats and 2nd endings, etc.), some less so (harmonics, stops in brass instruments, glissandi, tremolo, flutter tonguing, multiphonics, breath marks, turns, mordants, etc.). Furthermore, explicitly allowing for the great many rare cases only invites trouble from a recognition perspective, as these rare possibilities will inevitably be identified in unwanted locations. In this light, it is not reasonable to expect that an entirely automatic approach can "solve" OMR.

Commercial systems deal with this situation by isolating the recognition process, first performing black-box recognition, and then delivering the results for correction through a score-writing program such as *Sibelius* or *Finale*. In contrast, we (and other OMR researchers) believe a better solution involves the user as a *partner* in the recognition process. The user can "bless" correctly identified symbols, allowing the system to automatically improve its performance using these validated results as training examples. The user can identify the existence of symbols outside of the recognizer's core vocabulary, thus extending the vocabulary when appropriate while providing training examples. Additionally, the user can identify errors and allow the system to re-recognize subject to user-imposed constraints (e.g. this pixel lies within a note stem). Of course, in the most uncooperative cases the user can simply "tell" the computer the right answer, as with score-writing programs. Thus, building an interface that allows the user to participate in the recognition process is part of our challenge. During this last year we have created some simple IU tools in this direction, though most of this task still lies before us.

In short, we have a highly ambitious vision for OMR — far more than can be accomplished in a year or two. The remainder of this document describes the specific progress we have made over the course of the year. Some of the most important accomplishments are scientific or algorithmic in nature. In these cases we describe the essence of the results and why they are important, while referencing a more technical discussion. We will present both qualitative and quantitative recognition results. We also describe current and future directions.

2 Our State of the Art

Our vision for OMR is highly ambitions, including the development of core recognition technology for music, automatic adaptation of the system, developing the target symbolic representation, while integrating these elements into a system that allows a user to guide and correct the process in an efficient manner. While none of these tasks are "completed" — each one is a multi-year endeavor in its own right — we have made significant progress on the the core recognizer and its automatic adaptation, with some important development of ideas regarding symbolic representation, and some modest results toward a user interface. The remainder of this section explains our state of the art, highlighting the accomplishments over the last year of effort.

2.1 Page Structure

Before receiving this grant we had made significant progress in identifying the overall structure of a page of music. Similar structural decompositions form the first stage of every OMR approach we know. Our prior work in this area identifies the staves in a page of music, grouping these staves into systems, while

identifying measures for each system. While this work contains some original insights, it shares much with the many existing approaches, and is prerequisite for delving deeper into OMR. Though we have not tested this approach broadly yet, it performs well on the limited data we have tried. Consequently, we believe that structural identification is not the the most challenging aspect of OMR, while many approaches may produce satisfactory results here. In contrast, the core recognition engine, constituting our main focus under the period of this grant, is a highly challenging task with many potential pitfalls and chances for creative contribution.

2.2 Measure Recognition

2.2.1 Top-Down Recognition of Composite Symbols

Having located the measures through our structural decomposition, we proceed to identify the *contents* of these measures. We choose the measure as our fundamental unit of recognition since the interpretation of accidentals requires the entire measure context (accidentals carry through the measure), while the time signature constraint (the sum of note lengths in a measure equals the time signature) also figures prominently in our approach.

The focus for our core recognition engine is on the most essential musical symbols of the measure. We refer to these as the "what" symbols, expressing timing and pitch information (stems, note heads, beams, flags, accidentals, clefs, augmentation dots, ledger lines, rests, tuplet numbers, and ties). The remaining symbols are mostly concerned with the *manner* of performance (articulation, dynamics, text directions, (*de*)*crescendi*, etc.) though this distinction breaks down in some cases. Our approach builds specialized recognizers for the various objects we seek by capitalizing on the essential grammatical relations that give rise to the symbols' meaning. For instance, a beamed group must alternate between note stems and beams as one traverses the structure from left to right, with occasional partial beams interspersed. We begin by developing explicit grammars for the possible presentations of the "composite" symbols (isolated chords and beamed groups) which are formed of the "primitive"symbols of stems, note heads, flags, beams, and ledger lines. While a deeper discussion is beyond the scope of this document, such grammatical representations form the heart of the recognition strategies employed in speech and OCR by constraining the possible recognized results to configurations that make sense. For instance, flags must be "bound" to stems, which must be bound to note heads, while ledger lines can only appear in a well-defined arrangement for notes that lie off the staff. This measure analysis phase begins by identifying plausible *candidate* locations through inexpensive computations, then performing more thorough "grammatical"searches of these candidates, Thus, the search proceeds "bottom-up" — that is, by looking for composite symbols without yet formulating reasonable ways in which these symbols can fit together at, say, the measure level. The recognition literature includes a great deal of discussion on the relative virtues of bottom-up and "top-down" strategies, with our grammatical approaches falling into the latter category. In short, bottom-up schemes are more computationally tractable, while top-down approaches are more principled and function better when they are feasible.

A more scientific description of this work can be found in our recent paper: published in the proceedings of the *International Symposium on Music Information Retrieval* (ISMIR, 2011):

http://www.music.informatics.indiana.edu/papers/ismir11/ismir_omr.pdf.

2.2.2 Resolving Overlapping Configurations

The result of our initial bottom-up phase produces a collection of possible objects for each measure. Our principal concern in this phase is to identify nearly every existing core object in the measure, while we accept that this goal must also produce a number of unwanted objects as well. That is, we are willing to accept a number of "false positives" as long as the "false negatives" are rare. The result is a collection of overlapping object hypotheses that share "body parts" in mutually inconsistent ways. Our next phase seeks *variants* of these recognized objects such that the variants do not overlap; this may involve discarding some of the objects completely. We accomplish this goal by identifying the "regions of conflict," where several hypotheses lay claim to a certain portion of the image. For each such conflict we allow the relevant hypotheses to compete for the shared region. We re-recognize each hypothesis, subject to the constraint that it must avoid the region of conflict. We then seek the best-scoring collection of constrained and unconstrained hypotheses,

grouped in such a way that the final collection contains no overlap. A more detailed discussion of this work, as well as our core symbol recognizer, is presented in the paper referenced above.

2.2.3 Training

The biggest payoff for OMR will come from machine-printed notation, since such scores exist for a great variety of classical music while this version of the problem is much more tractable than its hand-written cousin. Machine-printed notation is characterized by a high degree of variation in the presentation of symbols *between* documents, with quite limited variation of symbols *within* a document. For example, there may be many different pixel configurations that correspond to a sharp sign throughout the entire IMSLP corpus, but within a single document one should expect usually minor variations on one or two basic templates. This allows the OMR system to "adapt" to the particular document at hand. Such automatic adaptation is commonplace for speech recognition systems, which can improve the recognition for a particular speaker using machine learning techniques. These tools should produce even better results with machine-printed OMR.

At present, our system has a simple-minded interface that allows the user to identify mistakenly recognized symbols by clicking on images such as those of Figure 1. The remaining symbols can be taken as correct examples, which our system uses to automatically adapt. We perform two kinds of training: parametric and template-based. Symbols such as note heads are represented either as single ellipses or, in the case of "whole" and "half" note heads, white ellipses within black ellipses. For objects such as these we have automatically learned the "parameters" of the ellipses, such as major and minor axes as well as orientation. For the remaining objects we identify prototypical "templates" that describe which pixels we expect to be black, white, or somewhere in between. Both of these approaches use standard machine learning techniques. A good part of our approach to training has already been implemented, though details remain to be filled in and extended to the many trainable parameters we encounter. At present we have automatically learned clefs, accidentals, note heads, rests, and ledger lines, though it is straightforward to extend these results.

2.2.4 Interpreting the Recognized Music

A final phase of measure recognition seeks to *interpret* the contents of each recognized measure, assigning actual pitches and rhythms to each recognized note, with rhythm providing the biggest challenge. In monophonic music, understanding the rhythm is straightforward when the recognition is correct: each note begins where its predecessor leaves off. However, rhythm interpretation becomes more complex when multiple voices appear in a single measure, as each note's predecessor is not as easy to identify. In this case one needs to partition the notes into voices, thus establishing monophonic streams of notes. Complicating voice partitioning, the number of voices commonly varies within the measure, as when two voices begin or end with a shared rest. Furthermore, rhythmic ambiguities also result from conventions regarding beamed groups in which it is common to leave out the "tuplet" numbers when the meaning is obvious (to a *human*). While this problem of rhythm interpretation, involving understanding of voices, clearly presents formidable obstacles, it must be solved if to understand the rhythmic context of a measure. However, there is further benefit, since the problem dovetails with the symbol recognition problem, discussed above, in an potentially fruitful manner, as follows.

After symbol recognition has been performed we are often left with multiple possible hypotheses for each identified symbol: perhaps a particular note may have one, two or no flags bound to it with scores given to these possible interpretations. We pose the rhythm interpretation problem as one that examines the notes of a measure, left to right (breaking ties arbitrarily), assigning the notes to voices, with a variable number of voices as the measure evolves. Using classic techniques from dynamic programming, it is possible to examine and score all possible such label sequences in a computationally tractable manner. While doing this, we consider multiple rhythmic interpretations for each note, weighted by their scores from the previous recognition phase. Dynamic programming then identifies the best-scoring interpretation that is consistent with the time signature. While this phase has been has been carefully formulated "on paper, we have just begun the actual implementation of this approach. Preliminary results will follow soon.

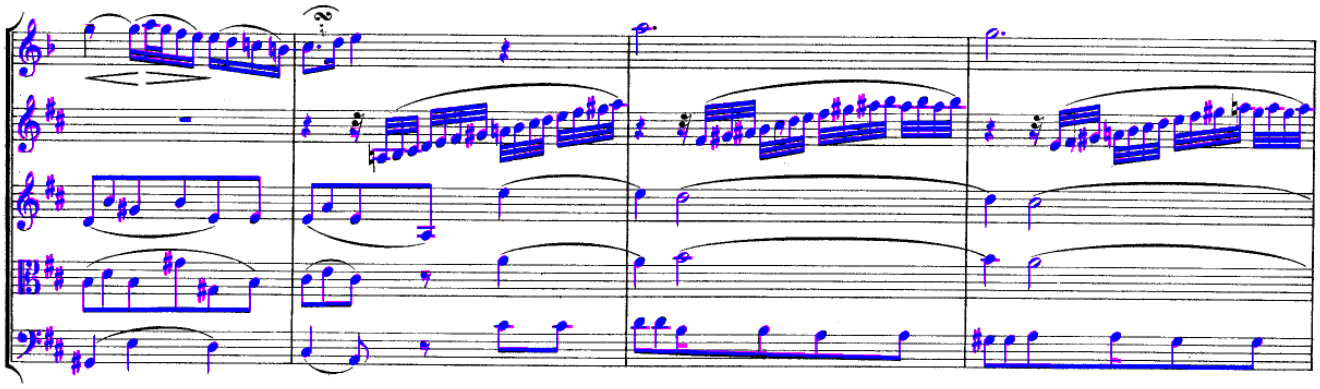

Figure 1: Example of recognition on page from Mozart Clarinet Quintet, K.581.

2.3 Evaluation

Here we first present a few examples and experimental results designed to convey our current level of success. Clearly we have both accomplished quite a bit and have a long way to go.

Figure 1 shows a system from the 2nd movement of the Mozart *Quintet for Clarinet and Strings*, K. 581, as recognized by our OMR program. This system has a number of complex beamed groups as well as variety of accidentals. The image has been colored so that blue denotes recognized "black" pixels, while red denotes recognized "white" pixels. Thus, any symbol colored as black has not been identified by our system, while any significant amount of red represents blank space that has incorrectly been recognized as something. It is easy to identify errors in this example, such as the missed 32nd rests (we don't yet even look for these), as well a couple of missed natural signs. We believe this example to be more or less representative of our current level of success, containing quite a bit that is correctly identified.

Figure 2 "zooms in" to show a number of errors in greater detail. These were chosen by hand as they represent *syndromes* of mistakes our recognizer makes, rather than isolated problems. A number of the errors are due to "out-of-vocabulary" problems, in which the actual symbol is not among those we currently seek. For instance, the example in the 3rd row and 1st column, (3,1), shows a double sharp that was misrecognized since we do not yet look for this unusual symbol. As slurs are also not yet in our vocabulary, the large amount of black ink they present tempts our recognizer to identify slurs as "known" objects, as in example in position (3,4). The best general antidote to this issue is to add the symbol in question to the vocabulary, since this allows for a competing hypothesis that will likely fit the image data better. However, in OMR one should always expect a certain number of out-of-vocabulary symbols, since, as discussed above, adding rare symbols to the recognizer will result in these symbols being identified in unwanted locations. Thus the majority of slurs that have *not* been recognized as something should be regarded as successes.

Example (4,2) shows a staff line that has been mistaken for a partial beam; this kind of mistake results from the difficulties of handling occlusion in recognition, since the domain of the partial beam is hidden by the staff line. We hope to correct this kind of error through better training of our models. Finally, example (4,3) shows a misrecognized augmentation dot; this happens frequently due to the smallness of these symbols, hence the "small sample size effect" familiar from statistics. Both of these latter mistakes will be addressed through the interpretation phase of our recognition procedure, discussed above but not yet implemented.

Table 2.3 gives a *numerical* description of our results on the first 5 pages from the Beethoven *Romance # 2 for Violin and Orchestra*. Pictures of the results in the manner shown above can be found at http://www.music.informatics.indiana.edu/papers/ismir11. In this experiment we hand-marked the pages identifying the locations of all primitive symbols in the image, using a simple program developed for this purpose by PhD student Jingya Wang. We then scored our recognition results for each primitive in terms of the "false positives" — the identified symbols that were not in fact true symbols, and the "false negatives" — the true symbols that were simply missed.

It should be stated that this kind of evaluation misses the mark in an important way since we are

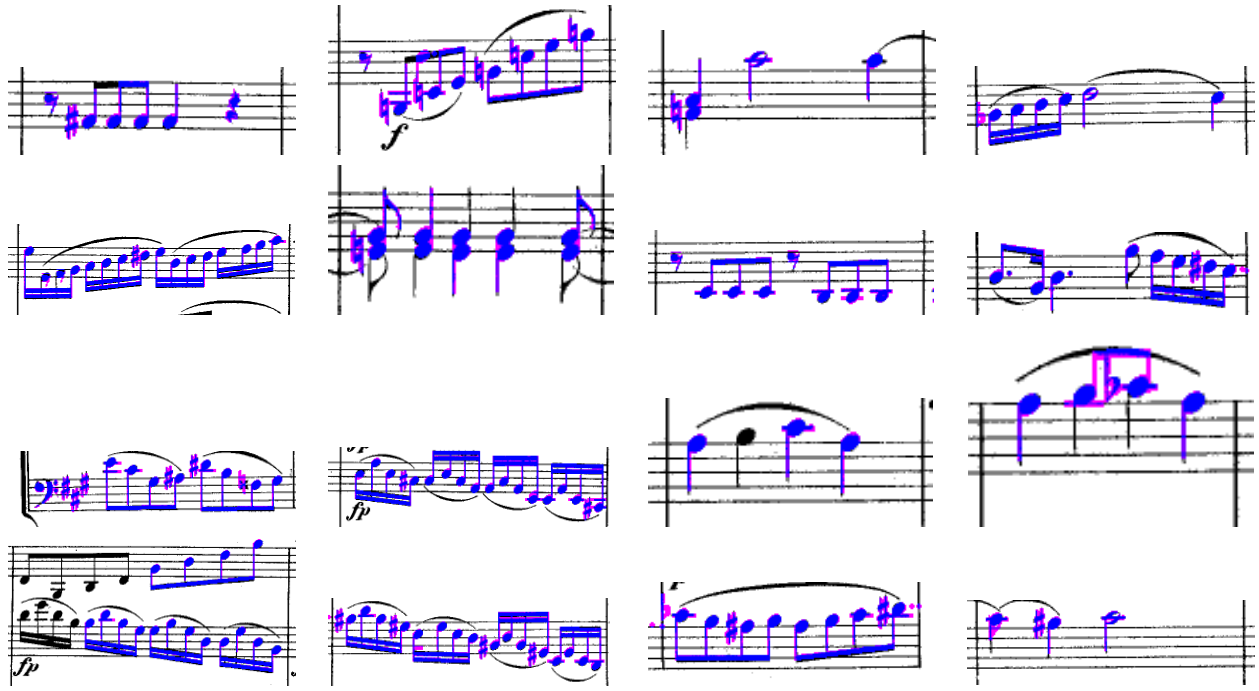

Figure 2: Examples of various misrecognized measures.

not measuring the accuracy of the resulting symbolic results — e.g. how many notes from the score do we recognize at the correct time with correct pitch, but rather the accuracy of an intermediate symbol recognition stage. The OMR literature contains quite a bit of discussion of this issue, and it is widely acknowledged that evaluation of fully symbolic output is complex without obvious solutions, or even good candidates. For instance, a missed clef change results in mistaken pitches for all subsequent notes in the voice, though it doesn't seem reasonable to evaluate in so harsh a manner. However, the intermediate evaluation we have performed is rather obviously related to the eventual symbolic results we will produce, and is simple (though labor-intensive) to perform and interpret.

It has been noted by many OMR authors that results can be very difficult to compare, as there is very little in the way of standard test sets with associated ground-truth. However, the false negative/positive error rates we present are in line with those reported in the academic literature for commercial systems.

3 Ongoing and Future Goals

The challenge posed by OMR is not one that can be solved in the space of a year or two. However, the potential benefits are significant, far-reaching, and lasting, more than justifying the decade-long effort we anticipate. What follows is a discussion of several aspects of OMR that fit into our ongoing vision, though remain undeveloped. We take this opportunity to formulate our thoughts and plans, though they remain subject to revision.

3.1 Symbolic Representation

The goal of OMR is, of course, to produce *symbolic* music representations — descriptions that express music in terms of notes with discrete pitch and length for each note, along with a wide variety of other information such as articulations, dynamics, repeat structure, ornaments, pedaling, etc. Symbolic music representations are somewhat well-studied, while the symbolic community offers several established and nearly complete encodings to choose from, such as MusicXML and the Music Encoding Initiative (MEI). Thus, it is not necessary to construct a new encoding scheme for OMR. Rather, the important issue is to decide what aspects of the music should be captured in the encoding. Naturally, this choice depends on

symbol name	**False +**		**False -**	
solid note head	.04	74/1724	.04	68/1718
note stem	.02	29/1573	.06	90/1634
ledger line	.07	51/701	.06	43/693
2 beam	.11	35/312	.04	13/290
1 beam	.23	76/331	.08	23/278
aug. dot	.52	252/481	.14	36/265
8th rest	.03	7/242	.04	10/245
3 beam	.04	6/138	.15	24/156
single flag down	.00	0/92	.36	51/143
whole rest	.21	28/132	.10	12/116
flat	.07	8/107	.05	5/104
quarter rest	.01	1/92	.10	10/101
open note head	.28	25/88	.29	26/89
single flag up	.02	1/50	.34	25/74
natural	.14	7/50	.30	18/61
treble clef	.00	0/60	.00	0/60
sharp	.36	21/58	.16	7/44
16th rest	.04	1/24	.21	6/29
bass clef	.00	0/20	.00	0/20
triple flag down	.43	9/21	.20	3/15
triple flag up	.59	13/22	.10	1/10
alto clef	.00	0/10	.00	0/10
4 beam	.33	1/3	.00	0/2
double flag up	-	0/0	1.00	1/1
double flag down	1.00	3/3	-	0/0
total	**.10**	648/6325	**.07**	472/6158

Figure 3: Error rates in terms of both "false positives" and "false negatives" for the symbols considered in the Beethoven Violin Romance #2 experiments.

the ease which which such aspects can be accurately understood from the document. Many fine details of symbolic representation, such as the beam grouping of notes and the directions of stems, come nearly for free from our OMR approach since we must recognize this detail to achieve the most basic understanding of the document's content. We plan to embed all such "free" information into our symbolic representation.

Whatever niceties one thinks a symbolic representation ought to have, the core pitch and rhythm content are certainly indispensable for most purposes; one cannot even play the music back without this understanding. Unfortunately, rhythm interpretation does *not* come for free from OMR, due to complexities involving voices and missing tuplet numbers. We discussed a strategy for understanding the rhythm of a measure in section 2.2, based on the idea of partitioning the notes into voices and using the time signature constraint to help choose among plausible hypotheses. This is our plan for recovering this vital rhythmic data. It should be mentioned that the identification of pitch is not completely straightforward, especially in large ensemble scores where one must know the transposition associated with each staff line.

Given the grand challenge presented by the OMR problem, it makes sense to pare the problem down whenever possible. Other than pitch and rhythm, we are inclined *not* to include any aspect of the score interpretation that do not come for free as a byproduct of OMR. An example of this is the distinction between slurs, ties, and phrase marks, all indicated with the same symbol, but whose meaning is often unclear and the subject of debate. Other examples would be the rhythmic interpretations of grace notes, when measured, or issues involving the execution of ornaments. As a general principle, we will choose not to interpret symbols except when necessary to achieve the most basic pitch/rhythm understanding of the music content. Refining this notion of "most basic" will depend on ongoing study, and the ease with which certain aspects of musical meaning can be easily recognized from the OMR results.

3.2 User Interface

A basic tenet of our approach, held from the onset of this project, is that recognition by itself will not be enough to "solve" the OMR problem. While the core recognition problem (the "what" symbols) is *somewhat* well-defined, this core has been extended in many different directions over several centuries. The result is the "heavy-tailed" nature of music notation, with a great many symbols that occur infrequently. Furthermore, from the vantage point of recognition science, it is mistake to try to continually broaden the range of special cases that the OMR system can accommodate — many of these situations are rare and would result in unwanted identification of rare symbols where they don't belong. (We really don't want to find mordants within *Tristan und Isolde*!) Our proposed solution is to limit the extent of what we try to accomplish with recognition alone, augmenting this by human input that corrects, directs, and otherwise aids the system.

At present we have only begun considering the way the user interface (UI) could combine human oversight with core recognition technology. For example, our current UI allows a user to delete various misrecognized symbols, simply by clicking on them, so they do not pollute the adaptation process with incorrectly labeled symbols. This is an example where it is particularly easy for the user to supply the needed input: "we want this flat here and we don't want this natural there." There are other examples where simple-minded UI approaches like this one may prove fruitful, while the demands of such cases can be satisfied by an "OMR editor" which allows the user to make such corrections. In essence, any isolated symbol or group of symbols whose meaning can be simply expressed by a user is a candidate for such an approach. For instance, the user could draw a bounding box around some text and type in its contents, or similarly identify the location of an accidental and label it by choosing from a menu. Given this user-supplied information, it is relatively easy to match the raw pixel data to the known interpretation, while this result can be displayed in the UI by coloring the recognized pixels as in Figure 1. The user would proceed in this way, both by looking at the recognized results and playing them back through editor until the result is satisfactory.

Approaches like are essential for cleaning up the detritus left by a recognition pass, but only go so far. In many cases, the information that recognition fails to provide is more complex than the isolation and labeling of a single symbol. Often the problem lies not in merely identifying a symbol or primitive, but also specifying relationships: this accidental belongs to this note head, which belongs to this stem, which belongs to this beam. We will explore the development of a UI that allows the user to express such relationships, though the result will be more difficult and time-consuming to user.

A better approach may be to allow the user to supply missing bits of information, while re-recognizing *subject to these user-generated constraints*. For instance, if the user identifies the correct label for a single

pixel, say as part of a triple flag, we can then reexamine the offending region in a way that is consistent that the user's labeling. One virtue of such an approach is that all of the internal relationships and dependencies that eventually give rise to the musical meaning are embedded in the resulting recognition. Another virtue is that the knowledge of the contents of one part of a measure can, as described in Section 2.2.4, be used to aid the recognition and interpretation of different parts of a measure. There seems to be considerable potential in such ideas, and certainly a great challenge in developing them.